

2021 秋《计算机硬件基础》

05 指令系统和汇编语言 作业

一、 填空

- 1、将 10 进制数 35 转换为 8 位二进制数是 0010 0011。
- 2、将二进制数 00010101 转换为 16 进制数是 0x15。
- 3、将 10 进制数-35 转换为 8 位二进制补码是 11011101。
- 4、将 10 进制数 204 转换为 8 进制数是 314。
- 5、请判断以下两个补码表示的二进制数做二进制加法后是否溢出： 溢出。

01101110

00011010

- 6、对 8 位 16 进制数 0x88 做符号扩展成 16 位数是：0x 0xff88。
- 7、下列代码段存储在内存中，起始地址为 0x00013244，分支指令执行后 PC 的两个可能的值分别是：0x 00013244 和 0x 00013254。同时，请在注释位置用伪代码形式对每条指令做出描述。

```
loop:lw$t0, 0($a0)    # to = mem[a0 + 0]  
      addi$a0, $a0, 4  # a0 = a0 + 4  
      andi$t1, $t0, 1  # t1 = t0 and 1  
      beqz$t1, loop    # if t1 != 0: goto loop
```

二、 简答与设计

- 1、将下列汇编语言指令翻译成机器语言代码，以 16 进制表示。

```
loop:addu$a0, $0, $t0    # 0x00082021  
      ori$v0, $0, 4      # 0x34020004  
      syscall            # 0x0000000c  
      addi$t0, $t0, -1    # 0x2108ffff  
      bnez$t0, loop       # 0x1500ffff  
      andi$s0, $s7, 0xffc0 # 0x32f0ffc0  
      or$a0, $t7, $s0     # 0x01f02025  
      sb$a0, 4($s6)       # 0xa2c40004
```

```
srl$s7, $s7, 4      # 0x0017b902
```

2、写一个 MIPS 汇编程序，要求对内存以“example100”为标签（label）的数据段中前 100 个字（words）的数据求和，并将结果存入紧跟在这 100 个字之后的内存中。

```
.data
example100: .word 4:100

.text
main:
    li $s0, 0
    li $t0, 0
    li $s1, 400

    FOR:
        beq $t0, $s1, END_FOR

        lw $t1, example100($t0)
        add $s0, $s0, $t1

        addi $t0, $t0, 4
        j FOR
    END_FOR:
        sw $s0, example100($t0)

        # li $v0, 1
        # move $a0, $s0
        # syscall

    li $v0, 10
    syscall
```

3、写一段 MIPS 汇编语言代码，将内存中“SRC”标签开始的 100 个字的一块数据转移到内存中另一块以“DEST”标签开始的空间中。

```

.data
SRC: .word 4:100
DEST: .space 400

.text
main:
    li $t0, 0
    li $s1, 400

    FOR:
        beq $t0, $s1, END_FOR

        lw $t1, SRC($t0)
        sw $t1, DEST($t0)

        # move $a0, $t1
        # li $v0, 1
        # syscall

        addi $t0, $t0, 4
        j FOR
    END_FOR:

    li $v0, 10
    syscall

```

4、写一个 MIPS 函数 lowbit，通过 \$a0 传入一个 32 位整数，将其在二进制表示下最低位的 1 及其后面的所有的 0 的二进制构成的数值再存入 \$a0。再写一段主程序，调用两次 lowbit 并输出结果，每次传给 lowbit 的数不同。

```

.data
newLine: .asciiz "\n"

.text
main:
    li $v0, 5
    syscall
    move $a0, $v0
    jal lowbit

```

```

    li $v0, 5
    syscall
    move $a0, $v0
    jal lowbit

li $v0, 10
syscall

lowbit:
    move $t0, $a0
    sub $t1, $zero, $a0
    and $a0, $t0, $t1
    li $v0, 1
    syscall

    li $v0, 4
    la $a0, newLine
    syscall

    jr $ra

```

5、写一个函数 FIB(N, &array)向内存中的一个数组（array）存入斐波那契数列的前 N 个元素。N 和 array 的地址分别通过 \$a0 和 \$a1 传递进来。斐波那契数列的前几个元素是：1, 1, 2, 3, 5, 8, 13,。

```

FIB:
    li $t0, 1 # i
    li $t1, 1 # last
    li $t2, 0 # last last

    li $t3, 1
    sw $t3, 0($a1)
    addi $a1, $a1, 4

    FOR:
        beq $t0, $a0, END_FOR

        add $t3, $t1, $t2
        sw $t3, 0($a1)

```

```

        move $t2, $t1
        move $t1, $t3

        addi $a1, $a1, 4
        addi $t0, $t0, 1
        j FOR
END_FOR:
        jr $ra

```

6、写一个函数，求出以“arr”为标签（label）的数据段中前 100 个字（words）的第二大字，并将结果存入紧跟在这 100 个字之后的内存中。

```

FIND_SECOND:
    li $s1, 0 # Max
    li $s0, 0 # subMax

    li $t0, 0
    li $s2, 100

    FOR:
        beq $t0, $s2, END_FOR

        sll $t2, $t0, 2
        lw $t3, arr($t2)

        bge $t3, $s1, GOTO_1
        bgt $t3, $s0, GOTO_2
        j PASS

    GOTO_1:
        move $s0, $s1
        move $s1, $t3
        j PASS

    GOTO_2:
        move $s0, $t3

    PASS:

        addi $t0, $t0, 1
        j FOR

```

END_FOR:

sll \$t2, \$t0, 2
sw \$s0, arr(\$t2)

move \$a0, \$s0
li \$v0, 1
syscall

jr \$ra